...e costruisci il tuo **LABORATORIO DIGITALE**

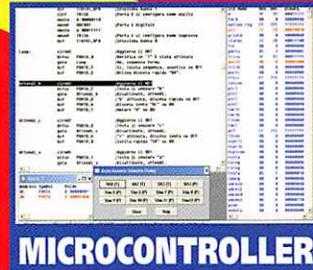
6,90 €

58

HARDWARE

DIGITALE DI BASE

MICROCONTROLLER

DIGITALE AVANZATO

TOTALMENTE PROGRAMMABILE!!!

9 771824 363008 41058

Peruzzo & C.

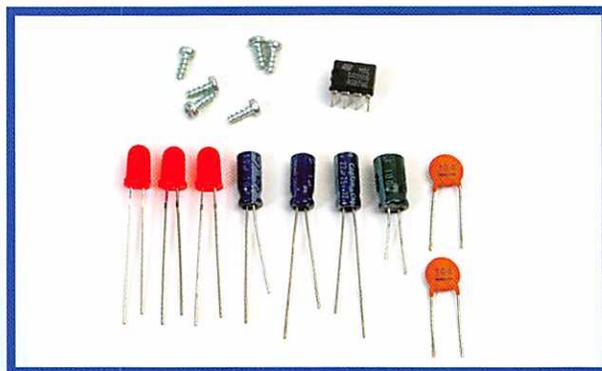
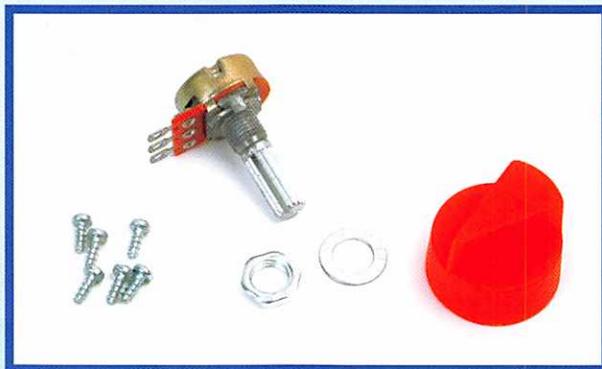
Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE"
si compone di
70 fascicoli settimanali
da suddividere
in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

IN REGALO in questo fascicolo

1 Circuito integrato 555
3 Diodi LED rossi
2 Condensatori 100 nF ceramici
1 Condensatore 1 µF elettrolitico
1 Condensatore 10 µF elettrolitico
1 Condensatore 100 µF elettrolitico
1 Condensatore 22 µF elettrolitico
6 Viti

IN REGALO nel prossimo fascicolo

1 Potenziometro con alberino metallico da 100 K Lin. con bullone e rondella
1 Manopola per potenziometro
6 Viti

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

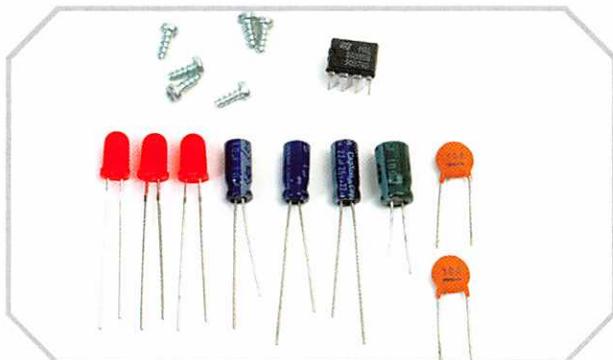
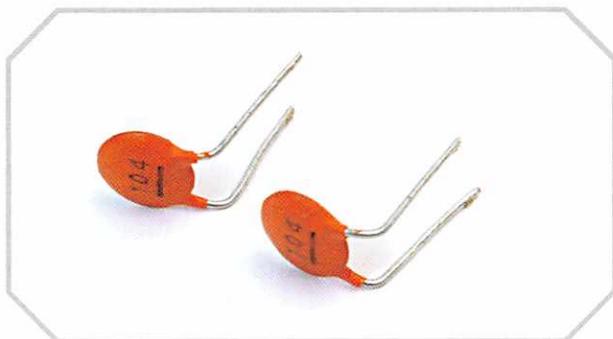
L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede. **Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail:** elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

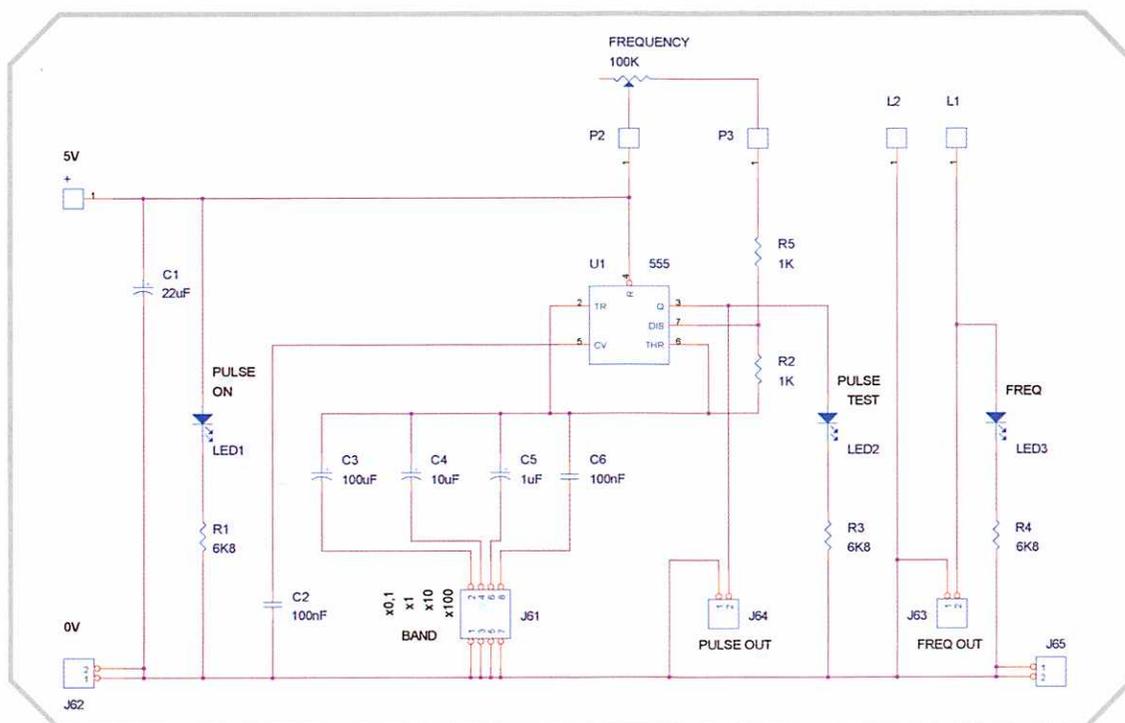
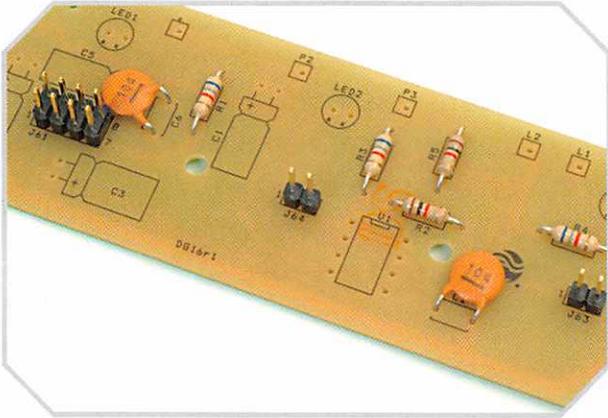
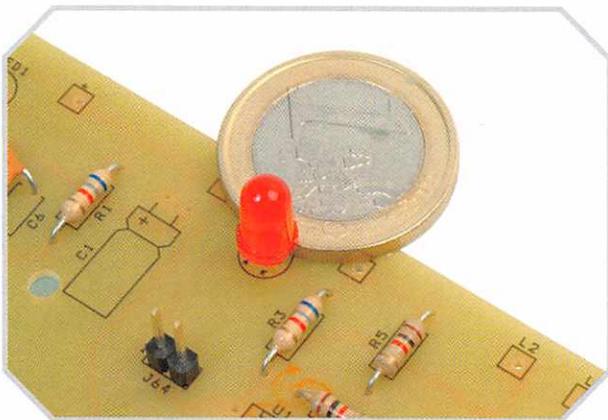
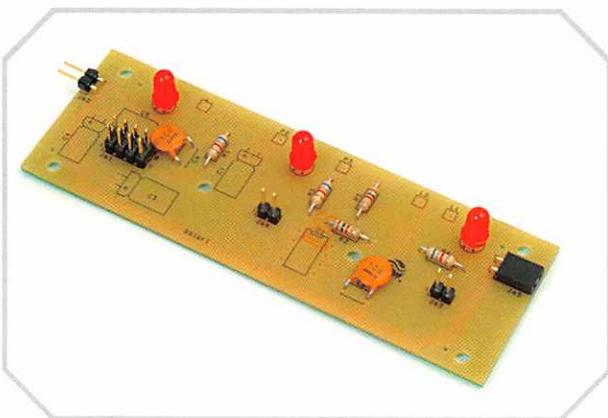
Microcontroller Esercizi con i microcontroller

Il generatore di impulsi

Componenti allegati a questo fascicolo.

Condensatori da 100 nF con i terminali piegati a 90°.

Con questo fascicolo vengono forniti i componenti necessari per completare la scheda DG16 che corrisponde al generatore di impulsi. Troverete il circuito integrato 555, 3 diodi LED di colore rosso, 6 condensatori e 6 viti.

Il montaggio di questo generatore di impulsi è facilitato dal fatto che il circuito stampato contiene molti degli elementi che andranno posizionati sul pannello frontale, quali i diodi LED e i connettori, riducendo in questo modo il cablaggio al collegamento del potenziometro e del positivo dell'alimentazione. Tuttavia il montaggio dei connettori e dei LED pone dei limiti all'altezza della scheda e questo ci obbliga a montare alcuni componenti in modo diverso rispetto agli abituali montaggi eseguiti finora.

I condensatori

I terminali dei condensatori con dielettrico ceramico da 100 nF devono essere piegati prima

Schema elettrico del generatore di impulsi.

I condensatori da 100 nF si montano piegati sulla scheda.

I LED devono rimanere sollevati almeno due millimetri dalla scheda.

Scheda DG16 con i tre LED montati.

dell'inserimento sulla scheda, quindi dovranno essere mantenuti coricati durante la saldatura finché questa non si sarà raffreddata. Dobbiamo evitare di fissarli direttamente con le dita per non correre il rischio di bruciarci col calore generato dalla saldatura. Nelle fotografie possiamo osservare il lato che deve essere piegato.

I LED

Questo circuito utilizza tre diodi LED da 5 mm di colore rosso, ognuno dei quali si monta inserito con i suoi terminali inseriti nei fori siglati con A e K. Se osservate il contenitore del LED vedrete che vicino alla base vi è una zona piatta, questa zona è la più vicina al terminale del catodo, il quale va inserito nel foro indicato con K; l'altro terminale, solitamente più lungo, viene inserito nel foro siglato con A.

Il contenitore del LED non deve rimanere appoggiato alla scheda ma a circa 2 mm dalla stessa.

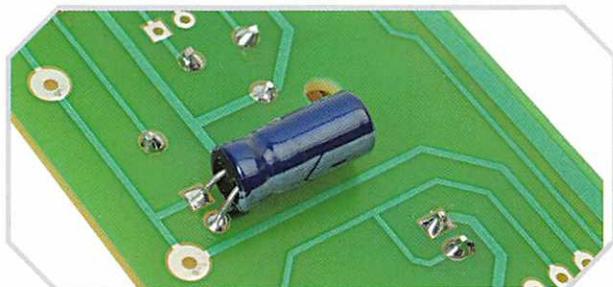
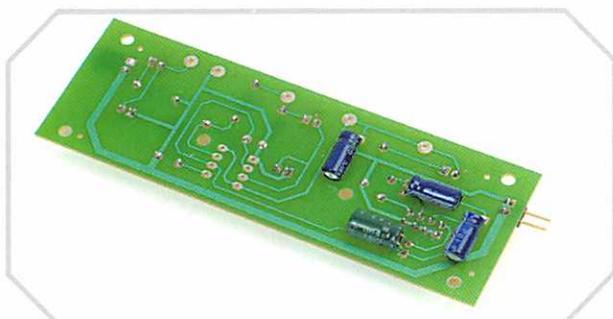
Per ottenere questa distanza il modo più semplice è quello di utilizzare come separatore una moneta da 1 euro o un'altra simile all'incirca della stessa misura.

Dopo aver montato il LED salderemo uno dei suoi terminali, toglieremo quindi la moneta, verificheremo che il LED sia perfettamente verticale e salderemo anche l'altro terminale.

Gli elettrolitici

In questa scheda, a causa della limitazione dell'altezza disponibile, i condensatori elettrolitici vengono montati dal lato delle saldature, inoltre devono essere piegati per facilitare la saldatura dei loro terminali.

È molto importante mantenere la polarità, dato che sono componenti polarizzati con un terminale positivo e uno negativo. Sul contenitore di ogni condensatore è presente normalmente una striscia verticale, argentata o di colore bianco, su cui abitualmente viene stampato il segno meno (-).

I quattro condensatori di questo fascicolo hanno valori diversi, quindi prima di saldare ognuno di essi dobbiamo verificare che sia

I condensatori elettrolitici si montano sull'altro lato della scheda.

I quattro condensatori elettrolitici montati.

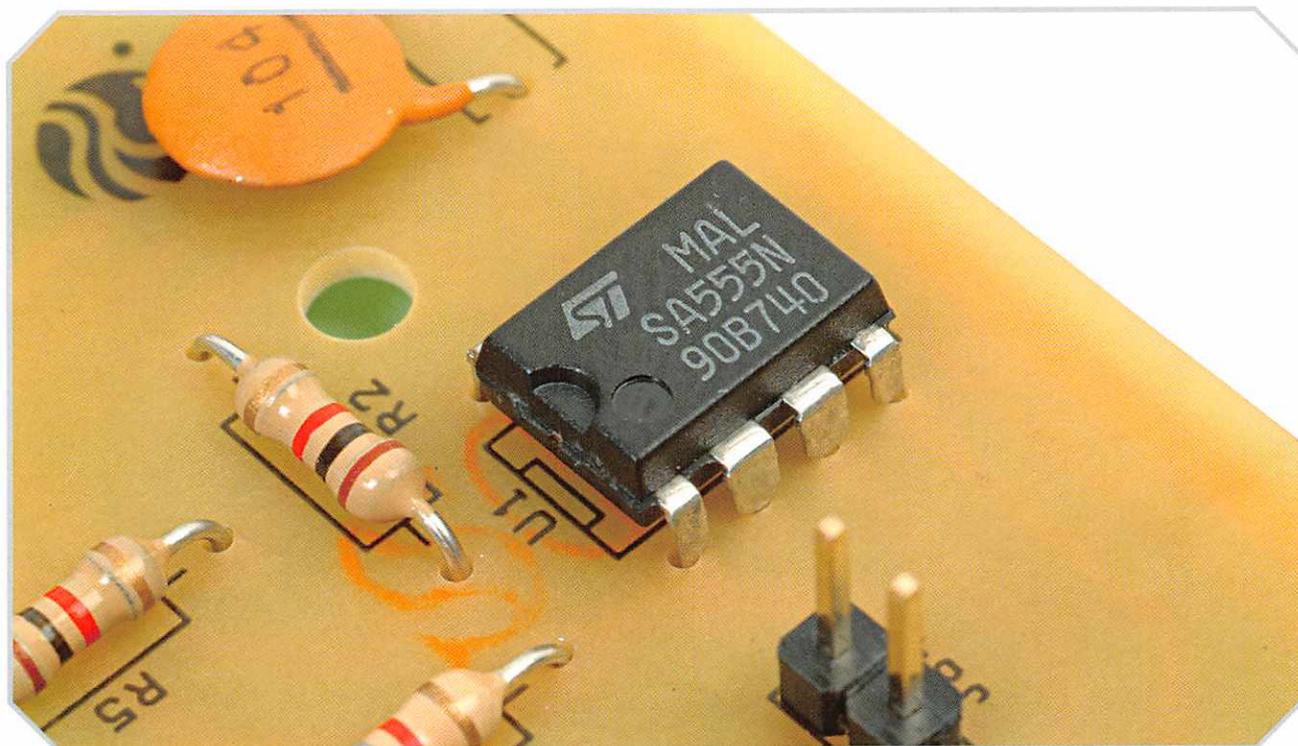
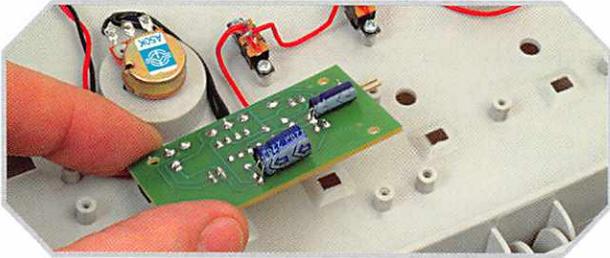
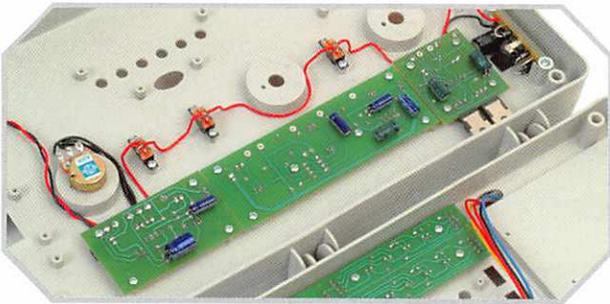
quello corretto e la sua polarità, controllando anche che sia ben piegato e aderente alla scheda prima di eseguire la saldatura.

Il 555

Il circuito integrato 555 ha otto terminali che vengono inseriti nella zona siglata con riferimento U1, bisogna però fare attenzione a inserirlo con la polarità corretta, quindi dobbiamo individuare sulla serigrafia una tacca che indica i terminali 1 e 8.

Sull'integrato il terminale 1 si identifica tramite una tacca di riferimento vicina a esso, solitamente è circolare e leggermente incassata.

Posizione della scheda DG16

Questa scheda si monta sul pannello superiore del laboratorio, tra DG15 e DG18. Dato che nel prossimo numero la dovremo nuovamente smontare per completare i collegamenti, è conveniente verificare se entra correttamente nel suo alloggiamento e fissarla con le sei viti ma senza stringere.

Dettaglio del circuito integrato 555.

Dobbiamo estrarre la scheda DG15.

Scheda DG16 nel suo alloggiamento.

Installazione

Anche se voi potete tranquillamente decidere di posticipare l'installazione della scheda al prossimo numero, descriviamo adesso i passaggi necessari per realizzare questa installazione.

Per prima cosa dobbiamo togliere le viti che fissano la scheda DG15, quella audio, al fine di poterla togliere. Faremo la stessa cosa con la DG18 e la solleveremo un po' per potervi collegare il connettore J62. Poi, con la scheda DG16 posizionata davanti al suo alloggiamento ma non ancora inserita, si collega il connettore della DG15 con J65 della DG16 e si abbassano le tre schede, avendo cura di far inserire i connettori e i LED nei fori del pannello a loro riservati.

Vi consigliamo di inserire alcuni ponticelli sui connettori i cui terminali dovranno fuoriuscire dal pannello frontale per fare in modo che, inserendo le viti, le schede rimangano centrate.

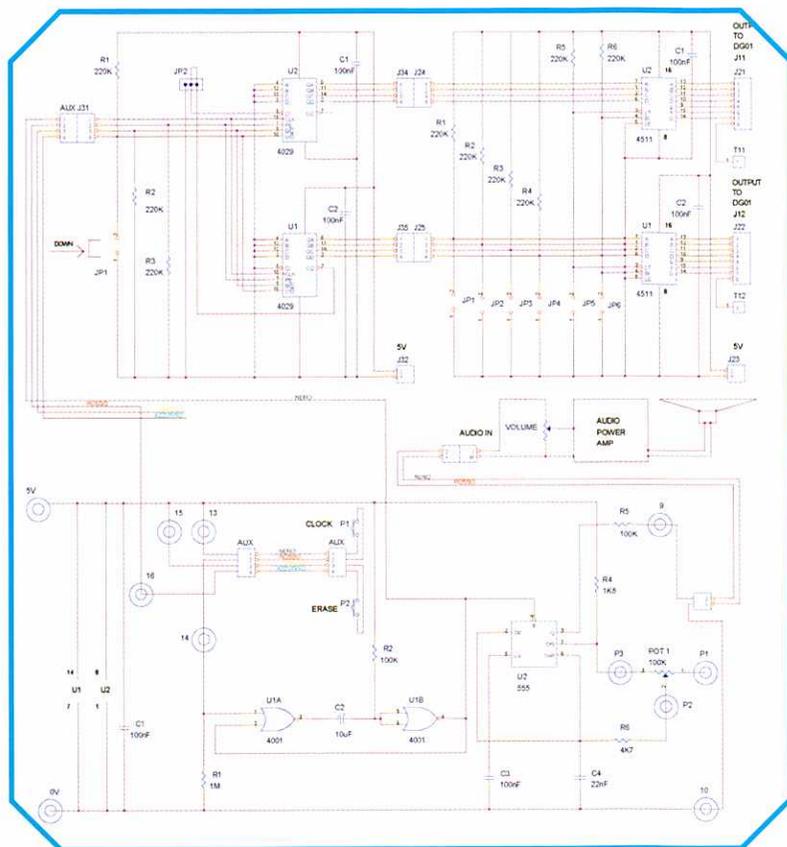
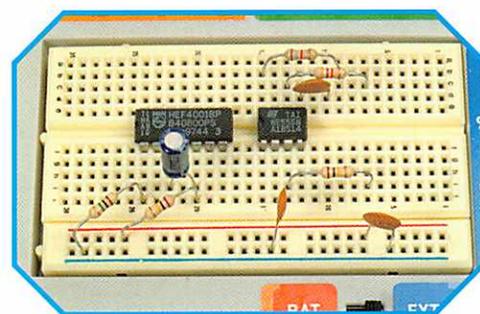
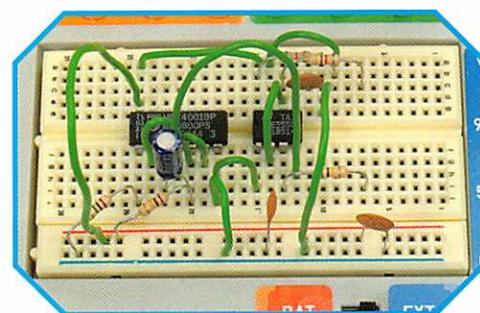
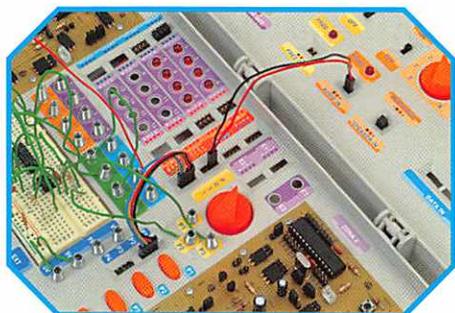
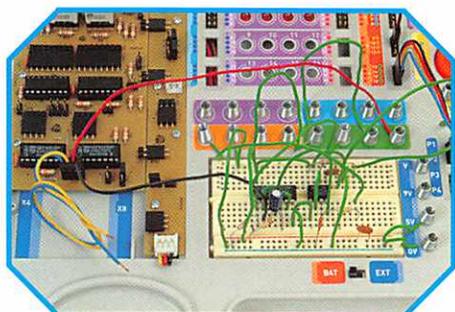
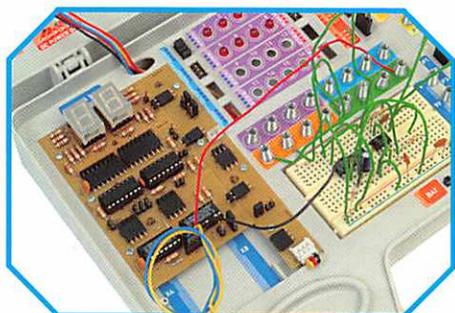
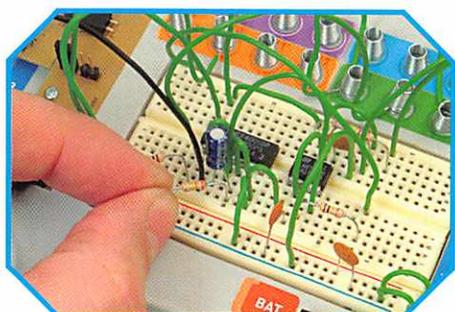
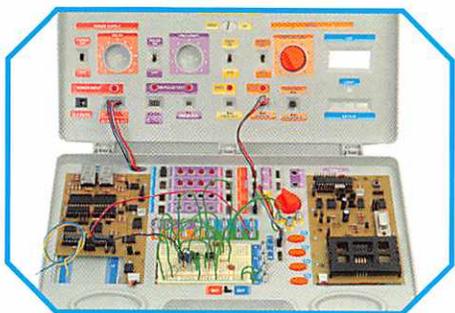
Pannello superiore del laboratorio.

Pulsante con avvisatore acustico

Questo circuito emette un breve segnale acustico ogni volta che si attiva il pulsante che fa avanzare il contatore. Questa applicazione ci ricorda i dispositivi utilizzati per organizzare le code di attesa nei centri commerciali o in alcuni uffici.

Il circuito

Osservando il circuito vedremo che è composto da parti che ci sono già note, nella parte superiore troviamo il circuito di controllo del contatore e in quella inferiore il resto del circuito. Le porte U1A e U1B formano un monostabile che si attiva con il pulsante P1, la cui uscita è utilizzata per far avanzare il contatore, applicando un impulso sull'ingresso del clock. La durata dell'impulso determina la durata dell'avviso, dato che si utilizza per abilitare il funzionamento del circuito del segnale acustico. Il segnale acustico è generato con U2, che è un 555 configurato come astabile, la frequenza del segnale di uscita di questo circuito si regola con POT1. Questo circuito si attiva quando il suo terminale passa a livello alto, e questo avviene quando l'uscita del monostabile è a livello alto.

Il segnale audio si applica all'ingresso dell'amplificatore audio dove riceve la potenza necessaria per pilotare l'altoparlante del laboratorio.

Schema del circuito.

Componenti sulla scheda Bread Board.

Cablaggio interno della scheda.

Collegamenti ai pulsanti e all'amplificatore audio.

Collegamenti al contatore.

Le schede DG04 e DG05 con i ponticelli predisposti per l'alimentazione a 5 V.

R2 determina la durata del segnale.

Esperimento completato.

Montaggio

Il montaggio di questo esercizio è facile nonostante lo schema risulti piuttosto grande, infatti il contatore e l'amplificatore audio sono montati in modo permanente sul laboratorio.

I collegamenti ai pulsanti P1 e P2, a cui sono assegnate le funzioni di avanzamento del contatore e di reset, si eseguono con un cavetto a quattro fili tra i connettori P1/P2 dei pulsanti e quelli numerati da 13 a 16 che terminano sulle quattro molle che portano lo stesso nome. I collegamenti di ingresso all'amplificatore si eseguono con un cavetto a due fili, rispettando però l'ordine di collegamento, senza scambiare i colori rosso e nero.

Il collegamento al contatore si esegue con un cavetto a quattro fili terminato con fili sciolti, dei quali utilizzeremo solamente il rosso e il nero; i ponticelli delle schede di alimentazione del contatore DG04 e DG05 devono essere configurati in modo che l'alimentazione sia da 5 V. L'alimentazione della scheda Bread Board deve essere da 5 V.

L'esperimento

Dopo aver verificato il circuito impostiamo al minimo il comando del volume dell'amplificatore audio e colleghiamo per prima cosa la sua alimentazione, AUDIO ON; poi collegheremo l'alimentazione al resto dei circuiti.

Premendo P1 avanzerà il contatore e udiremo un segnale acustico la cui durata può essere modificata variando il valore della resistenza scelta per R2, o scegliendo un altro valore per il condensatore C2. La frequenza del segnale di uscita si regola con POT1. Il contatore si imposta a zero con P2.

LISTA DEI COMPONENTI

U1	Circuito integrato 4001
U2	Circuito integrato 555
R1	Resistenza 1 M (marrone, nero, verde)
R2, R5	Resistenza 100 K (marrone, nero, giallo)
R4	Resistenza 1K8 (marrone, grigio, rosso)
R6	Resistenza 4K7 (giallo, viola, rosso)
C1, C3	Condensatore 100 nF
C2	Condensatore 10 µF elettrolitico
C4	Condensatore 22 nF

Esercizi con la scheda Smart Card

Abbiamo già parlato dei differenti utilizzi che si possono fare delle schede Smart Card. Ora vedremo come lavorare con esse per caricare nuovi programmi sul microcontroller.

Scrittura del Bootloader sul PIC

Quando si sviluppa un progetto si scrive il PIC con un programma e si monta un circuito elettrico. Se a questo circuito aggiungiamo un piccolo hardware basato su una Smart Card o su una comunicazione seriale, lasceremo aperta la possibilità di scrivere un nuovo programma sul PIC senza la necessità di estrarre quest'ultimo dalla sua sede o disporre di un hardware più completo per la scrittura. Per fare in modo che questo sia possibile, oltre alla circuiteria addizionale già citata, è necessario che sul PIC sia presente un Bootloader. Quando si avvia il PIC, questo programma cercherà di scrivere un nuovo programma di funzionamento e, nel caso non ne trovi nessuno (non ci sia la scheda Smart Card o non gliene venga fornito nessuno tramite la porta seriale del PC), farà partire quello che è presente nella memoria.

Per caricare il Bootloader sul PIC seguiremo la stessa procedura che abbiamo utilizzato per caricare un programma normale. Configureremo il Laboratorio per lavorare con IC-Prog, ovvero inseriremo i ponticelli su JP8 e JP9, sposteremo i ponticelli di JP1, JP2 e JP3 sulle posizioni 1 e 2 e collegheremo il cavo di trasferimento tra il PC e il Laboratorio.

Facciamo partire IC-Prog, configuriamo il dispositivo come PIC16F870 e apriamo il file "Bootloader.hex" presente sul primo CD allegato all'opera nella cartella Bootloader all'interno di Esercizi e Applicazioni.

Questo file si carica su IC-Prog con l'oscillatore, la protezione del codice e i bit della parola di configurazione già definiti. Questo perché all'interno del codice del programma sono stati definiti tali parametri. Potete vedere come fare questa definizione leggendo le informazioni allegate nella stessa cartella Bootloader.

Di seguito cancelleremo il contenuto del PIC, verificheremo e scriveremo il programma mediante l'opzione Programma Tutto.

Configurazione del Laboratorio per lavorare con IC-Prog.

Scrittura di un primo programma sulla scheda Smart Card

Dopo aver eseguito con successo la procedura precedente, dobbiamo caricare un pro-

Carichiamo Bootloader su IC-Prog.

Organizzazione del codice nella memoria di programma che deve essere corretta per lavorare con Bootloader.

gramma di funzionamento sulla scheda Smart Card.

Editeremo, per esempio, il file "ese1.asm", con il quale abbiamo già lavorato, e verificheremo che il codice occupi gli indirizzi di memoria successivi a 0xB0. Lavorando con Bootloader dobbiamo ricordare che questi occupa una precisa parte di memoria che dobbiamo evitare di sovrascrivere con i programmi di funzionamento, per questo il loro codice deve essere organizzato in modo da partire dall'indirizzo 0xB0.

Apriamo su IC-Prog il file "ese1.hex" e configuriamo il tipo di dispositivo, selezionando la memoria 24C16.

In questo caso non è necessario configurare i parametri dell'oscillatore, della protezione del codice e della parola di configurazione, dato che questi sono specifici dei microcontroller.

Osserviamo che caricando il file non cambia apparentemente nulla sulla videata di IC-Prog, in quanto gli indirizzi di memoria non hanno cambiato il loro valore. Ma se ci spostiamo verso altri indirizzi di memoria vedremo che il contenuto di alcuni di essi è cambiato.

Scrittura della scheda Smart Card

Torniamo a configurare i ponticelli del laboratorio (JP8 e JP9 estratti e JP1, JP2 e JP3 con i ponticelli su 2 e 3) per il funzionamento normale.

Configureremo i ponticelli JP4, JP5, JP6 e

Caricamento dell'esempio "ese1.hex".

Sulla scheda Smart Card caricheremo "ese2.hex".

Configurazione dei ponticelli sul laboratorio per il funzionamento normale.

Uniamo la porta di uscita con la matrice dei LED.

JP7 del Laboratorio sui terminali 1 e 2 per permettere la scrittura della scheda e inseriremo quest'ultima sullo zoccolo.

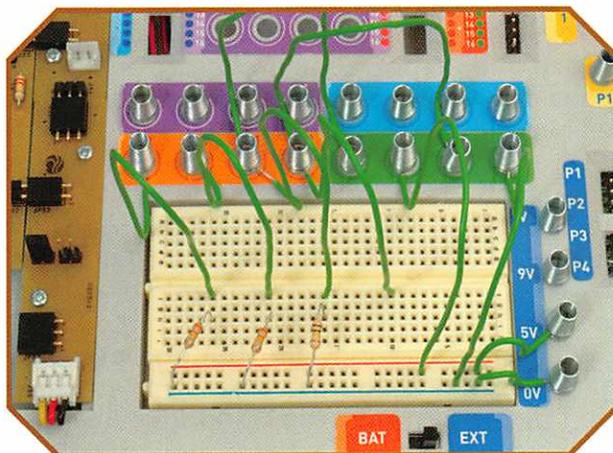
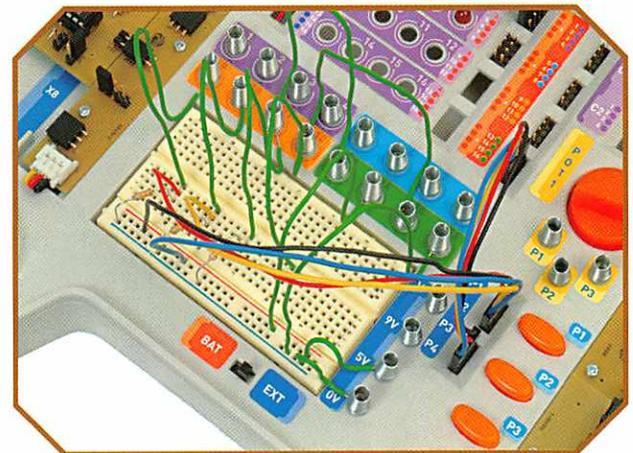
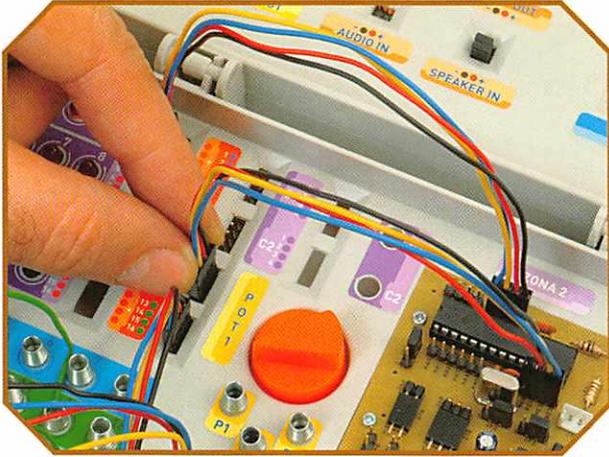
Continueremo selezionando l'opzione Programma Tutto e verificheremo che l'operazione sia stata eseguita correttamente leggendo il contenuto della memoria e confrontando quest'ultimo con il programma precedentemente inviato.

Per scrivere sulla scheda Smart Card un programma diverso si ripetono questi stessi passaggi, sarebbe utile però avere una scheda diversa per ogni programma, nel caso decidessimo di utilizzare sempre questa tecnica di programmazione, allo scopo di poter cambiare il programma del PIC in modo rapido e semplice. Dopo aver provato l'applicazione dell'esercizio ese1, caricheremo su IC-Prog l'esempio "ese2.hex".

Verificheremo, editando il codice, che anch'esso sia organizzato per funzionare con il Bootloader, ovvero che gli indirizzi che occupa nella memoria di programma inizino a partire da 0xB0, e lo scriveremo sulla Smart Card.

Montaggio del circuito elettrico

Prima di eseguire il montaggio dell'applicazione dobbiamo lasciare i ponticelli dei connettori JP4, JP5, JP6 e JP7 sulle posizioni 2 e 3 e togliere la scheda Smart Card dallo zoccolo.

Eseguiremo il montaggio elettrico che sarà comune per entrambi i circuiti. Mediante un cavetto uniremo la porta B di uscita (RB3:RB0)

Sulla scheda Bread Board eseguiremo il montaggio degli ingressi.

Inseriamo i pulsanti tra gli ingressi e il 5 V, tranne RC3 che andrà a 0 V.

Colleghiamo la porta di ingresso con le molle di collegamento.

con la matrice dei diodi LED. Ricordate che i ponticelli devono essere inseriti sui catodi dei diodi. Non utilizzeremo tutte le uscite perché sono sufficienti quattro di esse per osservare il funzionamento di questa applicazione.

Il montaggio degli ingressi si può fare utilizzando i pulsanti.

Come per le uscite, eseguiremo solamente il montaggio degli ingressi corrispondenti a RC3:RC0. Sulla scheda Bread Board inseriremo tre resistenze che da un lato verranno inserite sulle molle di collegamento arancio, e dall'altro a 0 V. Per l'ingresso RC3 eseguiremo un montaggio diverso, dato che non è necessaria la resistenza. Visto che la comunicazione con la scheda viene stabilita tramite RC3 e RC4, questi terminali non sono liberi come gli altri. Se osserviamo il circuito elettrico della scheda DG07 vedremo che questi terminali hanno una resistenza da 2k2 verso il positivo. Inseriremo i pulsanti tra il 5 V e il terminale di ingresso, mentre per il terminale RC3 il pulsante verrà inserito tra il terminale e 0 V.

Infine, collegheremo mediante un cavetto i terminali di ingresso RC3:RC0 al connettore corrispondente alle molle di collegamento 12:9 (colore arancio).

Proviamo i programmi attivando i pulsanti.

Prova di funzionamento

Quando collegheremo l'alimentazione al circuito apparentemente il microcontroller non farà nulla, non si attiverà nessuna uscita anche se proveremo tutte le combinazioni di ingresso. Internamente è in esecuzione il Bootloader ma, dato che non trova nessun programma da caricare, esso si ferma.

Se, a questo punto, inseriamo la scheda nel suo zoccolo e premiamo il pulsante di reset (SW2), o togliamo e ripristiniamo l'alimentazione, si riattiverà nuovamente il Bootloader sul PIC e cercherà un programma da caricare sul microcontroller dalle sue sorgenti esterne. Dato che la scheda Smart Card è stata inserita, la leggerà e scaricherà il programma che essa contiene. Questa operazione dura circa 15 secondi.

Trascorso questo tempo il microcontroller è pronto per lavorare con il nuovo programma e indica che la comunicazione è corretta perché i terminali RC3 e RC4 sono a livello alto (in questo caso possiamo vedere che il LED corrispondente a RC3 è sempre acceso).

Possiamo togliere la Smart Card dallo zoccolo in quanto potrebbe provocare delle anomalie di funzionamento essendo collegata alle linee RC3 e RC4 del PIC.

Se abbiamo caricato ese1 potremo vedere che lo stato delle uscite corrisponde allo stato degli ingressi. Quindi se azioniamo uno dei pulsanti il diodo LED corrispondente si accenderà (con RC3 attivando il pulsante il diodo si spegnerà).

Lavorando con ese2 (dovremo ripetere gli stessi passaggi eseguiti con ese1), se attiveremo i pulsanti P1 e P2 (ingressi RC0 e RC1) vedremo che le uscite si attiveranno in funzione della tabella corrispondente a ese2 e non come avveniva con ese1.

Controllo dei processi industriali: il trapano

L'impiego di microcontroller per il controllo di processi industriali è molto diffuso. La semplicità, la versatilità e il costo ridotto sono le caratteristiche che permettono alle applicazioni basate su microcontroller di essere la soluzione migliore per il controllo di molti processi industriali.

Macchina di foratura

L'applicazione che svilupperemo si prefigge di automatizzare una macchina di foratura. Questi tipi di processi normalmente richiedono la presenza di un operaio per l'esecuzione, sono monotoni e a volte anche pericolosi. Mediante l'integrazione di un semplice hardware basato su un microcontroller si semplifica il processo che viene eseguito automaticamente, ottimizzando tempi, qualità e costi.

Il processo che presenteremo di seguito è un processo reale di controllo di una macchina per foratura. Per forare un pezzo l'operaio deve avvicinare il trapano allo stesso, attivarlo, forare con la precisione necessaria e toglierlo, considerando così il pezzo terminato.

Processo industriale da automatizzare.

Sviluppo

Per automatizzare il processo posizioneremo due sensori, uno per rilevare l'inizio della foratura e l'altro per avvisare quando è stato realizzato il foro del pezzo. Inizieremo le operazioni mediante un pulsante e avviseremo acusticamente al termine del processo. Tutto questo si può pianificare nel seguente modo:

Azionando il pulsante "I" (RA0) la testa di foratura esegue una discesa rapida di avvicinamento, azionando il motore di discesa rapida "DR"(RB3).

Raggiunto il sensore "b" (RA2), si attiva il relè "M" (RB1) che fa girare il motore della punta e si esegue una discesa lenta per la foratura del pezzo "DL" (RB4). Quando si attiva il sensore "c" (RA3) si considera che il pezzo

Controllo automatico di una macchina per foratura.

Ingressi	RA0	*Pulsante che indica l'inizio del processo, pezzo nuovo.*
	RA1	*Finecorsa che indica che il trapano si trova nella posizione di origine.*
	RA2	*Sensore per indicare l'inizio della rotazione del trapano.*
	RA3	*Sensore di pezzo forato.*
Uscite	RB0	*Attiva la salita rapida del trapano.*
	RB1	*Attiva il relè che attiva il motore di rotazione del trapano.*
	RB2	*Segnale acustico di fine del processo.*
	RB3	*Attiva il motore di discesa rapida del trapano.*
	RB4	*Attiva la discesa lenta per forare il pezzo.*

Segnali di ingresso e uscita.

```
TRAPANO - Blocco note
File  Modifica  Formato  Visualizza  ?
;        ESERCIZIO: Macchina da foratura
;       ---------------------------------
;
;Azionando il pulsante "I" (RA0) la testa di foratura esegue una discesa rapida di avvicinamento,
;attivando il motore di discesa rapida "DR" (RB3). Raggiunto il sensore "b" (RA2), si attiva
;il relé "M" (RB1) che fa girare il motore della punta e viene eseguita una discesa lenta per
;la foratura del pezzo "DL" (RB4). L'attivazione del sensore "c" (RA3) segnala che il
;pezzo é stato forato. Inizia quindi una salita rapida della testa di foratura "SR" (RB0)
;mentre il relé "M" (RB1) di rotazione rimane attivato. Quando si raggiunge il termine della corsa "a"
;(RA1), si ferma la salita rapida "SR"(RB0), il relé di rotazione "M" (RB1) e si attiva un
;segnale acustico "A" (RB2) di avviso. Il ciclo inizia nuovamente con l'attivazione del pulsante "I" (RA0).

                List    p=16F870        ;Tipo di processore
                include "P16F870.INC"   ;Definizione dei registri interni

                org     0x00            ;Vector di Reset
                goto    Inizio

                org     0x05            ;Salva il vector di interrupt

Inizio          clrf    PORTB           ;Cancella i latch di uscita
                bsf     STATUS,RP0      ;Seleziona banco 1
                clrf    TRISB           ;Porta B si configura come uscita
                movlw   b'00000110'
                movwf   ADCON1          ;Porta A digitale
                movlw   b'00011111'
                movwf   TRISA           ;Porta A si configura come ingresso
                bcf     STATUS,RP0      ;Seleziona banco 0
```

Configurazione delle porte di I/O.

sia forato. Si inizia una salita rapida della punta "SR" (RB0) e durante questo tempo il relè "M" (RB1) di rotazione continua a essere attivato. Quando viene raggiunto il finecorsa "a" (RA1), si ferma la salita rapida "SR" (RB0), il relè di rotazione "M" (RB1) e si attiva un segnale acustico "A" (RB2). Il ciclo comincia con una nuova attivazione di "I" (RA0).

Abbiamo definito i dispositivi necessari per automatizzare il processo e, allo stesso tempo, abbiamo stabilito gli ingressi e le uscite con cui lavorerà il microprocessore. Non sono necessari altri dispositivi, dato che per risolvere questa applicazione è possibile utilizzare un microcontroller più semplice, cosa che ridurrebbe maggiormente i costi.

Codice

Predisporremo ora una soluzione utilizzando il nostro PIC16F870. Dopo aver definito tutti i passaggi e averli inseriti in un organigramma, sarà sufficiente far riferimento a esso e sviluppare il codice mantenendo l'ordine indicato.

Intesteremo il programma e definiremo i dispositivi con cui lavorare, in questo caso le porte di ingresso e di uscita. Utilizzeremo la porta B per le uscite e la porta A per gli ingressi, specificando che questi ultimi saranno digitali. Il programma principale resterà fermo nell'attesa che arrivi il segnale di ingresso RA0, segnale che indica l'inizio del processo. Se si attiva, spegneremo il segnalatore acustico (RB2 a 0) e abbasseremo il trapano (RB3 a 1). Manterremo questi segnali fino a quando il sensore ci indicherà che abbiamo raggiunto il punto in cui il trapano deve iniziare a funzionare, nel momento in cui il sensore si attiva (RA2 a 1), abbasseremo il trapano più lentamente (RB3 a 0 e RB4 a 1) e questo inizierà a funzionare (RB1 a 1). Il sensore "b" ci indicherà che il pezzo è stato forato correttamente (RA3 a 1), e a questo punto ritireremo il trapano facendolo risalire rapidamente (RB4 a 0 e RB0 a 1). Quando raggiunge il finecorsa (RA1 a 1) scollegheremo i segnali di movimento del trapano e attiveremo il segnale acustico generando l'avviso di pezzo terminato.

Organigramma dell'applicazione.

```
Loop:           clrwdt                      ;Aggiorna il WDT
                btfss   PORTA,0             ;Verifica se "I" è stato attivato
                goto    Loop                ;No, sequenza ferma
                bcf     PORTB,2             ;Si, inizia sequenza, acustico su OFF
                bsf     PORTB,3             ;Attiva discesa rapida "DR".

Attendi_b       clrwdt                      ;Aggiorna il WDT
                btfss   PORTA,2             ;Testa il sensore "b"
                goto    Attendi_b           ;Disattivato, attendi.
                bcf     PORTB,3             ;"b" attivato, discesa rapida su OFF
                bsf     PORTB,4             ;Discesa lenta "DL" su ON
                bsf     PORTB,1             ;Motore "M" su ON

Attendi_c       clrwdt                      ;Aggiorna il WDT
                btfss   PORTA,3             ;Testa il sensore "c"
                goto    Attendi_c           ;Disattivato, attendi.
                bcf     PORTB,4             ;"c" attivato, discesa lenta su OFF
                bsf     PORTB,0             ;Salita rapida "SR" su ON

Attendi_a       clrwdt                      ;Aggiorna il WDT
                btfss   PORTA,1             ;Testa il sensore "a"
                goto    Attendi_a           ;Disattivato, attendi.
                bsf     PORTB,2             ;"a" attivato, acustico su ON
                bcf     PORTB,0             ;Salita rapida su OFF
                bcf     PORTB,1             ;Motore su OFF
                goto    Loop

                end                         ;Fine del programma sorgente
```

Codice che risolve l'applicazione.

Compilazione

Apriremo MPLAB per procedere alla compilazione e simulazione del programma, creando un nuovo progetto a cui aggiungeremo il codice realizzato; selezioneremo poi l'opzione Build All.

Il codice si deve compilare senza errori, come possiamo verificare nella figura.

Simulazione

Quando realizziamo un progetto per automatizzare un processo industriale, prima di implementarlo fisicamente, dobbiamo verificare che risponda correttamente. Non è consigliabile fermare la produzione di un'azienda per montare un sistema che poi non funziona. Per evitare difetti di funzionamento, simuleremo prima il programma.

In MPLAB apriremo le finestre dei registri delle funzioni speciali, e una finestra per poter osservare unicamente i registri di ingressi e uscite (PORTA e PORTB). Vedremo questi registri in binario, perché in questo modo potremo verificare la risposta di ogni pin in modo indipendente. Ricordate che per cambiare la visualizzazione bisogna accedere alle proprietà e selezionare Binary, dopo aver selezionato il registro che si vuole vedere nella finestra.

Per simulare il pin di ingresso apriremo il simulatore di stimoli asincroni e assegneremo i cinque ingressi ai pulsanti corrispondenti.

Inizieremo la simulazione passo a passo e vedremo come il programma ripete un ciclo in attesa che l'ingresso si attivi. Imposteremo tutti gli stimoli dei segnali di ingresso a livello alto (High) e simuleremo il processo. Inizieremo cliccando con il mouse sul pulsante assegnato al pin RA0, per simulare l'attivazione dell'ingresso; se proseguiamo nell'esecuzione vedremo che il programma esce dal ciclo ed esegue i primi due ordini di uscita. Ripeteremo questa operazione nel resto del programma, ovvero attiveremo gli ingressi che simulano i sensori e i finecorsa per uscire dai cicli di attesa e osserveremo nella finestra dei registri di uscita in binario che le uscite si attivano correttamente.

Se si desidera fare un nuovo pezzo dovremo impostare i pin del simulatore di stimoli a livello basso (Low) e ripetere nuovamente tutti i passaggi.

Conclusioni

Il programma ha risposto correttamente alla simulazione, quindi lo potremo scrivere sul microcontroller e iniziare a realizzare l'implementazione fisica del sistema.

È indispensabile eseguire la simulazione dei

```
Building TRAPANO.HEX...

Compiling TRAPANO.ASM:
Command line: "C:\PROGRA~1\MPLAB\MPASMWIN.EXE /p16F870 /q C:\PROGRA~1\MPLAB\PROGETTI\TRAPANO.ASM"
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TRAPANO.ASM 23 : Register in operand not in bank 0.  Ensur
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TRAPANO.ASM 25 : Register in operand not in bank 0.  Ensur
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\TRAPANO.ASM 27 : Register in operand not in bank 0.  Ensur

Build completed successfully.
```

Risultato della compilazione.

Finestra per vedere in binario le porte di I/O.

Asynchronous Stimulus Dialog

RA0 (P)	RA1 (P)	RA2 (P)	RA3 (P)
...	Stim 6 (P)	Stim 7 (P)	Stim 8 (P)
...	Stim 10 (P)	Stim 11 (P)	Stim12 (P)

Assign Pin...
✔ Pulse
Low
High
Toggle
Help...

Close Help

Forziamo gli ingressi con il simulatore di stimoli.

programmi che sono destinati a processi industriali, dato che qualsiasi errore può far fermare la produzione, il ché comporterebbe una perdita economica considerevole.

Molte delle applicazioni per le quali si utilizza un microcontroller semplificano molto i processi industriali e non richiedono una programmazione complessa. Grazie a questo esempio reale avete potuto verificare come un microcontroller si possa convertire in un elemento fondamentale per un processo produttivo.

MPLAB IDE - C:\PROGRA~1\MPLAB\PROGETTI\TRAPANO.PJT

File Project Edit Debug PICSTART Plus Options Tools Window Help

c:\progra~1\mplab\progetti\trapano.asm

```
            bsf     STATUS,RP0      ;Seleziona banco 1
            clrf    TRISB           ;Porta B si configura come uscita
            movlw   b'00000110'
            movwf   ADCON1          ;Porta A digitale
            movlw   b'00011111'
            movwf   TRISA           ;Porta A si configura come ingresso
            bcf     STATUS,RP0      ;Seleziona banco 0

Loop:       clrwdt                  ;Aggiorna il WDT
            btfss   PORTA,0         ;Verifica se "I" è stato attivato
            goto    Loop            ;No, sequenza ferma
            bcf     PORTB,2         ;Si, inizia sequenza, acustico su OFF
            bsf     PORTB,3         ;Attiva discesa rapida "DR".

Attendi_b   clrwdt                  ;Aggiorna il WDT
            btfss   PORTA,2         ;Testa il sensore "b"
            goto    Attendi_b       ;Disattivato, attendi.
            bcf     PORTB,3         ;"b" attivato, discesa rapida su OFF
            bsf     PORTB,4         ;Discesa lenta "DL" su ON
            bsf     PORTB,1         ;Motore "M" su ON

Attendi_c   clrwdt                  ;Aggiorna il WDT
            btfss   PORTA,3         ;Testa il sensore "c"
            goto    Attendi_c       ;Disattivato, attendi.
            bcf     PORTB,4         ;"c" attivato, discesa lenta su OFF
            bsf     PORTB,0         ;Salita rapida "SR" su ON

Attendi_a   clrwdt                  ;Aggiorna il WDT
            btfss   PORTA,1         ;Testa il sensore "a"
            goto    Attendi_a       ;Disattivato, attendi.
```

Special Function Register Win...

SFR Name	Hex	Dec	Binary
w	1F	31	00011111
tmr0	00	0	00000000
option_reg	FF	255	11111111
pcl	12	18	00010010
pclath	00	0	00000000
status	1C	28	00011100
fsr	00	0	00000000
porta	01	1	00000001
trisa	1F	31	00011111
portb	08	8	00001000
trisb	00	0	00000000
portc	00	0	00000000
trisc	FF	255	11111111
intcon	00	0	00000000
pir1	00	0	00000000
pie1	00	0	00000000
pir2	00	0	00000000
pie2	00	0	00000000
tmr1l	00	0	00000000
tmr1h	00	0	00000000
t1con	00	0	00000000
tmr2	00	0	00000000
pr2	FF	255	11111111
t2con	00	0	00000000
ccpr1l	00	0	00000000
ccpr1h	00	0	00000000
ccp1con	00	0	00000000
rcsta	00	0	00000000
txreg	00	0	00000000
rcreg	00	0	00000000
txsta	02	2	00000010
spbrg	00	0	00000000
adresh	00	0	00000000
adresl	00	0	00000000
adcon0	00	0	00000000
adcon1	06	6	00000110
pcon	00	0	00000000
eedata	00	0	00000000

Watch_1

Address	Symbol	Value
05	PORTA	B'00000001'
06	PORTB	B'00001000'

Asynchronous Stimulus Dialog

RA0 (T)	RA1 (T)	RA2 (T)	RA3 (P)
Stim 5 (P)	Stim 6 (P)	Stim 7 (P)	Stim 8 (P)
Stim 9 (P)	Stim 10 (P)	Stim 11 (P)	Stim12 (P)

Close Help

Ln 36 Col 1 | 57 | WR No Wrap INS | PIC16F870 | pc:0x12 | w:0x1f | --Z dc c | Bk On | Sim | 4 MHz | Proj

Videata di MPLAB durante la simulazione.